

예제로 쉽게 배우는

아두이노

김진환 · 장성용 지음

1.1 Arduino란?

1.2 Arduino의 하드웨어

1.3 Arduino의 소프트웨어

1.4 Arduino IDE 준비하기

1.5 Arduino IDE 사용하기

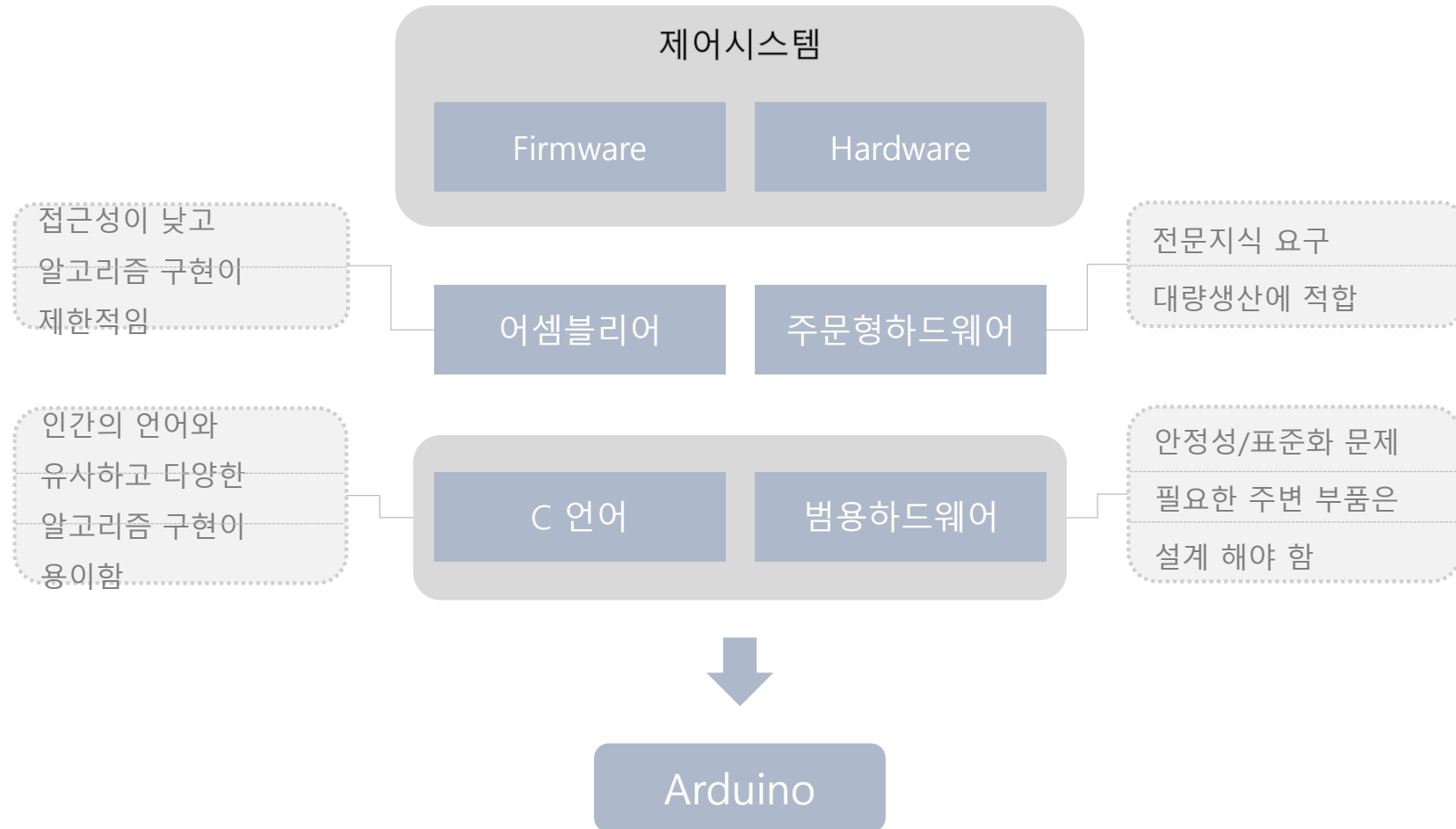
1.6 스케치의 기본 구성

1.7 전압, 전류, 저항

1.8 브레드 보드

1.9 Arduino UNO Starter kit

1.1 Arduino란?



1.1 Arduino란?

2005년 Italy의 Massimo Banzi & David Cuatielles에 의해 개발

예술가
취미생활
학생

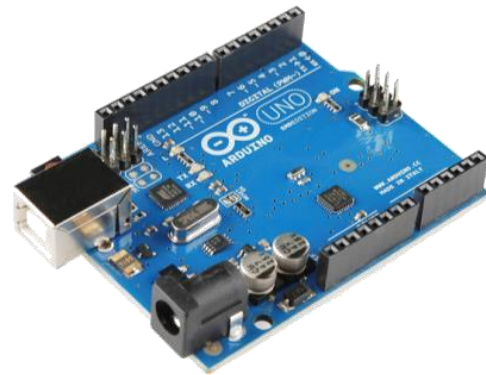
전자공학
교육

누구나 쉽게
사용 가능한
제어장치

오픈소스
하드웨어

GSM Wifi
Ethernet
Motor drive
등의 실드 제
공

다양한
라이브러리



LabView

MATLAB

AVR Studio

범용
하드웨어
의
표준

1.2 Arduino의 하드웨어

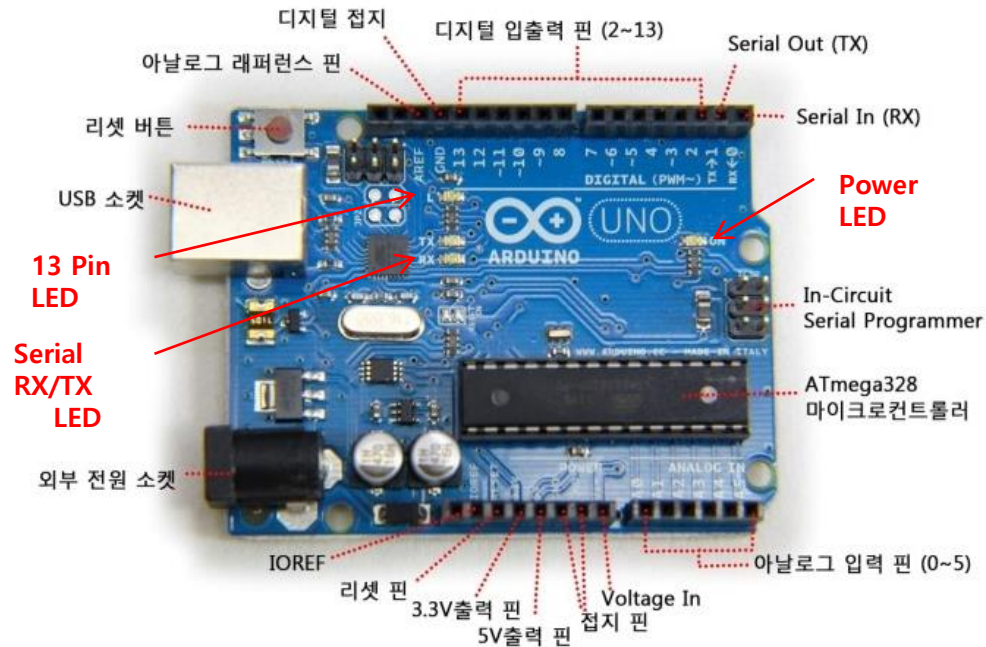
Arduino Board



Shield (통신모듈)



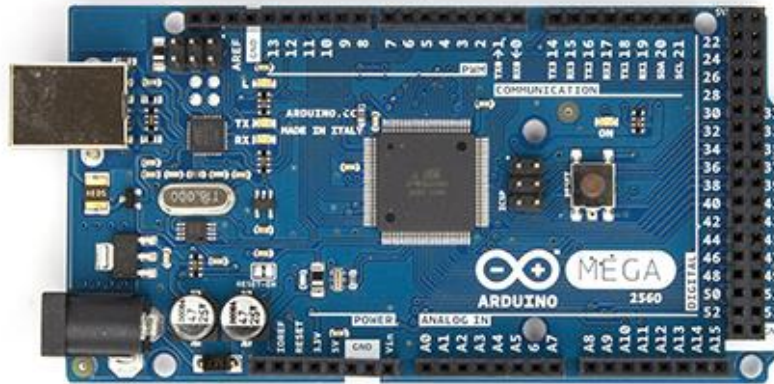
1.2 Arduino의 하드웨어



✓ Arduino UNO R3

- ATmega328 microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 6 Analog Inputs
- 32KB Flash Memory
- 16Mhz Clock Speed

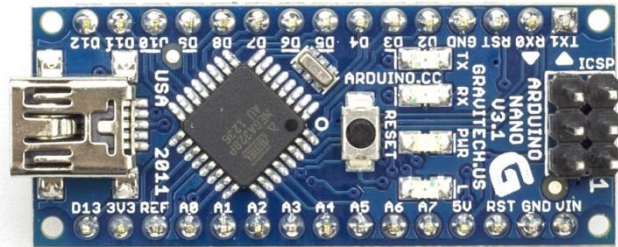
1.2 Arduino의 하드웨어



✓ Arduino MEGA 2560

- ATmega2560 microcontroller
- Input voltage: 7~12V
- 54 Digital I/O Pins (6 PWM outputs)
- 16 Analog Inputs
- 4 UARTs
- 256KB Flash Memory
- 16Mhz Clock Speed

1.2 Arduino의 하드웨어



✓ Arduino Pro NANO

- ATmega168/328 microcontroller
- Input voltage: 7~12V
- 14 Digital I/O Pins (6 PWM outputs)
- 8 Analog Inputs
- 16KB Flash Memory
- 16Mhz Clock Speed

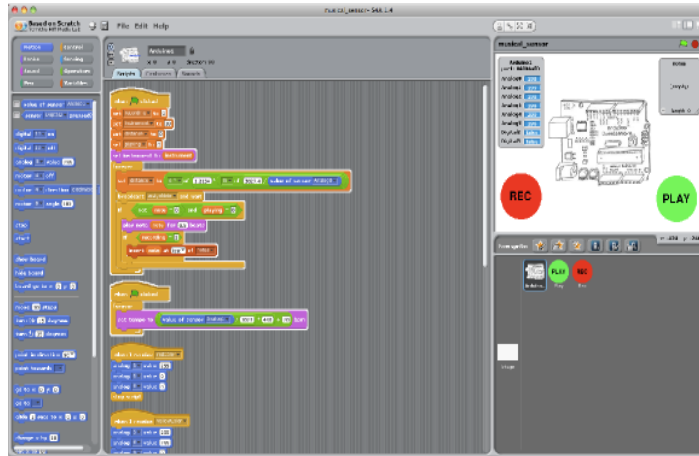
1.3 Arduino의 소프트웨어

스크래치

- ✓ MIT에서 만든 그래픽 언어
- ✓ 교육용, 알고리즘 구현에 제한적

C 언어

- ✓ 모든 컴퓨터 시스템에서 사용할 수 있는 프로그래밍 언어

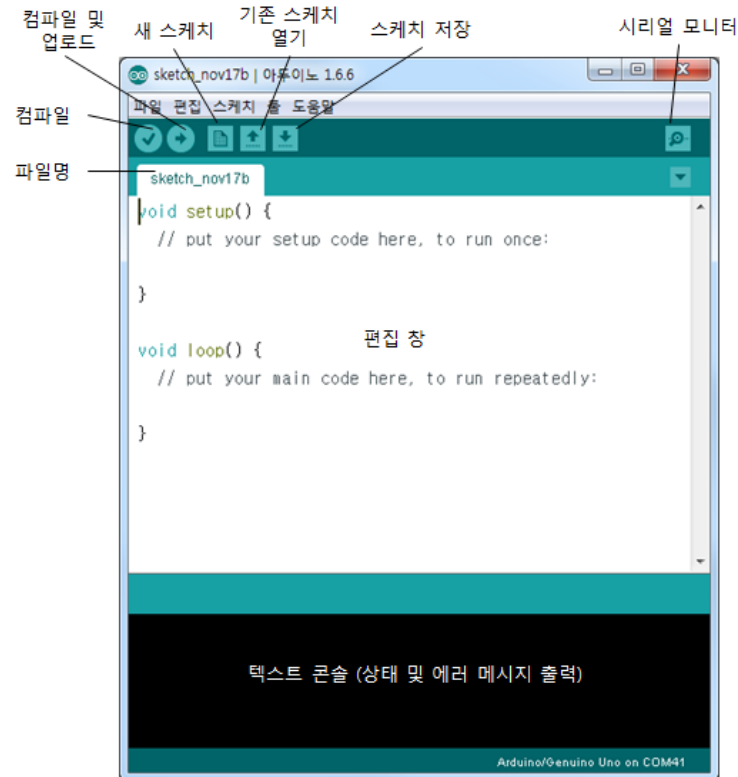


1.4 Arduino IDE 준비하기

Arduino
IDE
준비하기

14 ~ 16 페이지의 Step1 ~ Step9를
참고하여 Arduino IDE를 설치해 보자.

1.5 Arduino IDE 사용하기



파일명: 파일명은 새 파일을 생성시키면 우선 오늘의 날짜로 표시된다. 이후에 저장 버튼을 누를 때 재설정할 수 있다.

컴파일: 작성한 스케치를 Arduino가 인식할 수 있는 언어로 변경해준다. 단순히 스케치의 오류를 확인하거나 Arduino에 업로드할 필요가 없을 때 컴파일만 실행시킨다.

컴파일 및 업로드: 컴파일과 Arduino로의 업로드를 연속적으로 수행한다. 컴파일에서 에러가 없으면 자동으로 업로드된다.

새 스케치: 새로운 스케치를 작성한다. 기존의 편집 창은 유지한 채 새로운 창이 열린다.

기존 스케치 열기: 저장되어 있던 스케치 파일을 연다.

스케치 저장: 현재의 스케치를 저장한다.

시리얼 모니터: 시리얼 통신 상태를 볼 수 있는 창을 연다. Windows의 하이퍼터미널과 유사하다.

편집 창: 스케치를 작성하고 편집한다.

텍스트 콘솔: Arduino의 상태 및 에러 메시지를 출력한다.

1.6 스케치의 기본 구성



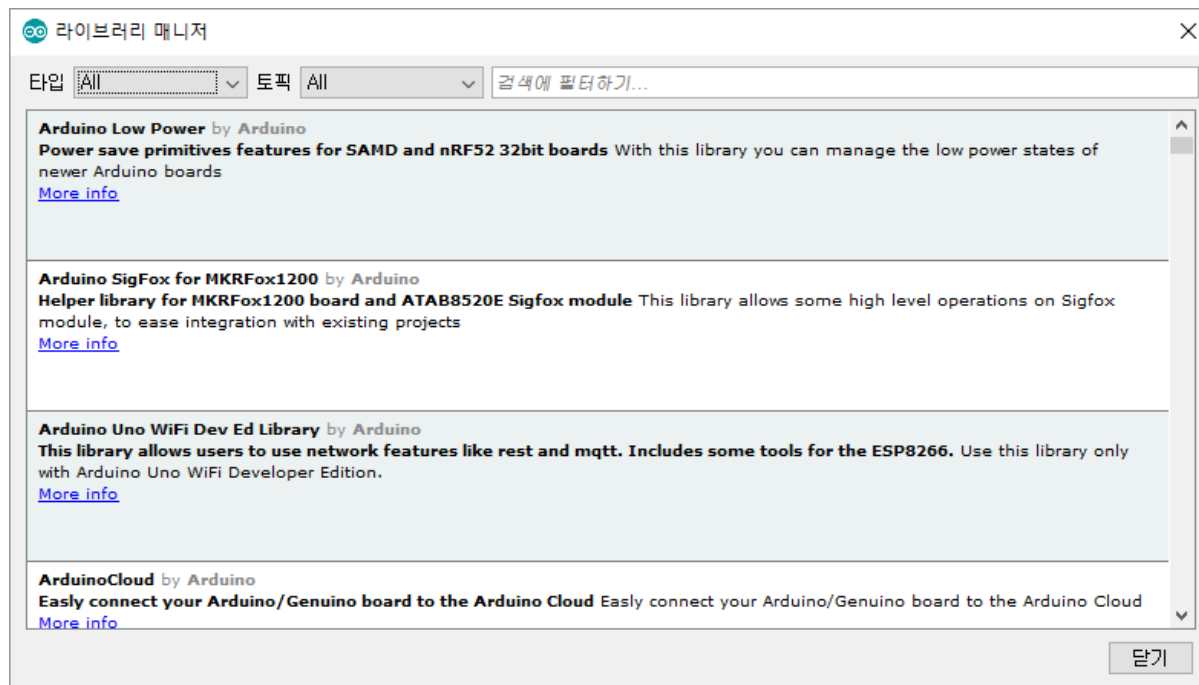
주석: 스케치의 이름, 목적, 제작자, 제작일, 갱신일, 갱신 내용 등의 정보를 적는다. 특히 갱신일과 갱신한 내용은 프로그램을 작성하는데 있어서 매우 중요한 일이므로 항상 정확히 기재하는 습관을 들이자.

변수, 상수 라이브러리 등의 정의: 스케치에서 사용할 전역 변수, 상수, 라이브러리 등 사전 설정이 필요한 것들을 정의한다.

설정 루틴 (setup): 설정루틴에서는 각 핀의 인풋과 아웃풋 및 시리얼 통신 등을 설정해준다. Arduino가 동작할 때 최초 한 번만 실행하여야 하는 명령어를 넣기도 한다.

반복 루틴 (loop): 무한 반복되는 루틴으로 C언어의 'main()' 함수와 동일한 기능을 한다. 동작에 관한 스케치는 이 부분에서 작성한다.

1.6 스케치의 기본 구성



‘스케치->라이브러리포함하기->라이브러리관리’ 메뉴로 가면 그림1.11과 같이 라이브러리를 관리할 수 있는 메뉴로 들어갈 수 있다. 예제에 필요한 라이브러리는 이곳에서 간편하게 받아 사용할 수 있다.

1.7 전압, 전류, 저항

전압
[V]

- ✓ 전위가 높은 쪽과 낮은 쪽의 차이
- ✓ 1쿨롱(coulomb: 전하의 단위)의 전하 갖고있는 에너지
- ✓ Arduino에서는 직류 3.3[V]와 5[V]를 지원

전류
[A]

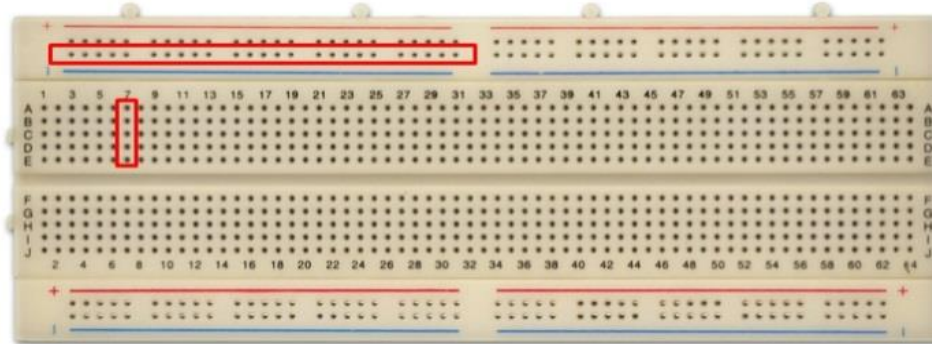
- ✓ 1초당 1쿨롱의 전하가 단위 면적을 통과했을 때를 1[A]로 정의
- ✓ Arduino에서는 1/1000[A] 단위인 [mA]를 사용

저항
[Ω]

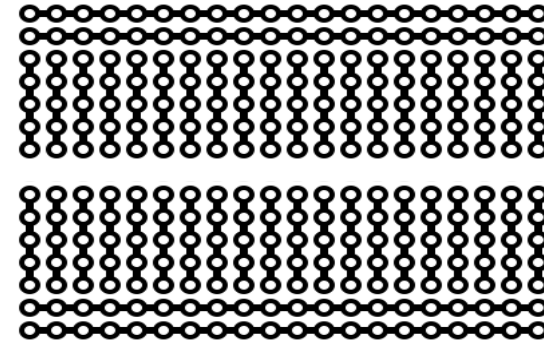
- ✓ 전류의 흐름을 방해하는 정도를 나타냄
- ✓ 색 띠나 숫자로 값을 표시
- ✓ Arduino에서는 칩 (chip) 형태의 저항이 사용

1.8 브레드 보드

시제품 제작이나 실험용 와이어를 보드에 꽂아 사용



빨간색 묶음 홀끼리 내부회로가 연결되어 있음



내부 결선